mn

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/660,166	09/11/2003	Paul Reuben Day	ROC920030145US1	7590

30206 7590 03/05/2007
IBM CORPORATION
ROCHESTER IP LAW DEPT. 917
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER	
PHAM, MICHAEL	
ART UNIT	PAPER NUMBER
2167	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/05/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 10/660,166	**Applicant(s)** DAY ET AL.	
	Examiner Michael D. Pham	**Art Unit** 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 February 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-16 and 18-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-16 and 18-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *12/20/06*.
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Detailed Action

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 12/20/06 has been entered.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 3-16, 18-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Application Publication 2004/0210563 by Zait et. al. (hereafter Zait) further in view of U.A. Patent 6502089 by Amundsen et. al. (hereafter Amundsen).

Claim 1:

Zait discloses, "a method for monitoring a query during runtime, said query involving a plurality of join operations; the method comprising the steps of:"

"running the query according to a first join order" [0024 and 0003, query is executed according to the one or more operations in an execution plan. Operations may consist of a join order. 0004, discloses determining an optimal join order therefore there must exist other join orders and the most optimal one is selected.];

"concurrent with running the query, collecting performance statistics about each of the join operations" [0024, statistics information is collected from operation which may include join operations.];

"changing the first join order, during running of the query, to a second join order based on the statistics" [0034, The plurality of collected execution statistics are then used to improve performance of the query statement can be improve the performance of the query statement. Performance of the query statement may include altering the execution plan (i.e. changing a first join order to a second join order based on statistics is possible.).]

Zait does not explicitly disclose

"generating a first portion of a result set for the query while running the query according to the first join order";

; and

"generating a second portion of the result set for the query while running the query according to the second join order".

On the other hand, Amundsen discloses if a multi-way join is requested by the user, e.g. a first relation is to be joined on a given attribute with a second relationis to be joined on a given attribute with a third relation, it is useful to know the join fanout (i.e. number of intermediate results that will be produced by, the join of the first and second relations as compared to the join fanout of the join of the second and third relations) (col. 23 lines 17-21). When processing a compound sql query, it is most efficient, generally speaking, to perform first those parts of the query that produce the smallest solution set of tuples, because doing so minimizes the number of intermediate results that must be produced, stored , and then processed in later parts of the query (col. 23 lines 25-30). Col. 13 l. 39-43, the user query specifies multiple join operations, and statistics are generated for two or more join operations and used to determine an order in which said join operations are performed, e.g. the join operation with the lowest statistic is performed first. Hence, Amundsen suggests generating a first portion of a result set for the query while running the query according to the first join order, and further suggests generating a second portion of the result set for the query while running the query according to the second join order. That is, as Amundsen states, it is most efficient, generally speaking, to perform first those parts of the query that produce the smallest solution set of tuples (i.e. generating a first portion of a result set for the query while running the query according to the first join order), and because Amundsen discloses a multi-join operations of course, generating a second portion of the result set for the query while running the query according to the second join order is further suggested.

Both Amundsen's and Zait's system disclose systems that further optimize queries, and therefore are both within the same field of endeavor. For the above reasons, it would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Amundsen's teachings of performing first those parts of the query that produce the smallest solution set of the tuples to Zait's system for minimizing the number of intermediate results that must be produced, stored, and then processed in later parts of the query.

Claim 3:

As to claim 3, Zait and Amundsen disclose the claimed limitation subject matter in claim 1, Zait further discloses "collecting additional statistics about each of the join operations after the first join order is changed to the second join order" (0029, snapshot is taken after execution of the selected operation.).

Claim 4:

As to claim 4, Zait and Amundsen disclose the claimed limitation subject matter in claim 3, Amundsen further disclosing "changing the second join order to either the first join order or a third join order based on the additional statistics" (col. 13 lines 41-43, statistics are generated for two or more join operations, and used to determine an order in which said join operations are performed.).

Claim 5:

As to claim 5, Zait and Amundsen disclose the claimed limitation subject matter in claim 1, Amundsen further disclosing "a first join that includes a first table and a second table; a second join that includes the first table and a third table" (col. 4 lines 40-51, discloses example of choosing between joining a name/address table with a city table first then joining the population table, or to join the city table with the population table before joining the name/address table.).

Claim 6:

As to claim 5, Zait and Amundsen disclose the claimed limitation subject matter in claim 1, Amundsen further disclosing "determining respective fan-in statistics for the first join and second join" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.)"; and" "changing the first join order to a second join order if the respective fan-in statistics indicate that the second join is more likely to cause fan-in than the first join." (col. 13 lines 40-43, multiple join operations. Join operation with the lowest statistic is performed first.)

Claim 7:

As to claim 7, Zait and Amundsen disclose the claimed limitation of claim 5, Amundsen further disclosing "determining respective fan-out statistics for the first join and the second join" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.); and "changing the first join order to a second join order if the respective fan-out statistics indicate that the second join is less likely to cause fan-out than the first join." (col. 13 lines 40-43, multiple join operations. Join operation with the lowest statistic is performed first.)

Claim 8:

As to claim 8, Zait and Amundsen disclose the claimed limitation of claim 5, Amundsen further disclosing "determining respective fan-in statistics for the first join and second join" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.); "determining respective fan-out statistics for the first join and the second join" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.); and "changing the first join order to a second join order based on a combination of the respective fan-in and fan-out statistics." (col. 13 lines 40-43, multiple join operations. Join operation with the lowest statistic is performed first.)

Claim 9:

As to claim 9, Zait and Amundsen disclose the claimed limitation of claim 1, Zait further discloses, "identifying a predetermined sample size" [0032, sample of execution statistics]; "performing the step of collecting statistics for the predetermined sample size" [0032]; "evaluating the collected statistics" [0032]; and "changing the first join order to a second join order based on the collected statistics." [0034].

Claim 10:

As to claim 10, Zait and Amundsen disclose the claimed limitation of claim 1, Zait further discoses "collecting additional statistics for substantially all of the query" [0032-0033]; "comparing the additional statistics with the collected statistics" [0032-0033]; and "adjusting the predetermined sample size, for use by a subsequent query, according to results of the comparing step"[0032-0033].

Claim 11:

As to claim 11, Zait and Amundsen disclose the claimed limitation of claim 1, Zait further discloses "running an other query after the query" [0024, essentially after finishing a query another query must run]; and "selecting an initial join order for the other query based on the collected performance statistics" [0024, selects an execution plan for new query.].

Claim 12:

Zait discloses "A method for optimizing a query join order during runtime, said query involving a plurality of join operations, the method comprising the steps of:"

"running the query according to a first join order" [0024 and 0003, query is executed according to the one or more operations in an execution plan. Operations may consist of a join order. 0004, discloses determining an optimal join order therefore there must exist other join orders and the most optimal one is selected.];

“concurrent with running the query, collecting performance statistics about each of the join operations” [0024, statistics information is collected from operation which may include join operations.];

“based on the collected statistics, selecting a preferred join order, while running the query, such that the query continues to run according to the preferred join order” [0034, The plurality of collected execution statistics are then used to improve performance of the query statement can be improve the performance of the query statement. Performance of the query statement may include altering the execution plan (i.e. changing a first join order to a preferred join order based on statistics is possible.).].

Zait does not explicitly disclose

“generating a first portion of a result set for the query while running the query according to the first join order” and

“generating a second portion of the result set for the query while running the query according to the preferred join order”.

On the other hand, Amundsen discloses if a multi-way join is requested by the user, e.g. a first relation is to be joined on a given attribute with a second relations to be joined on a given attribute with a third relation, it is useful to know the join fanout (i.e. number of intermediate results that will be produced by, the join of the first and second relations as compared to the join fanout of the join of the second and third relations) (col. 23 lines 17-21). When processing a compound sql query, it is most efficient, generally speaking, to perform first those parts of the query that produce the smallest solution set of tuples, because doing so minimizes the number of intermediate results that must be produced, stored , and then processed in later parts of the query (col. 23 lines 25-30). Col. 13 l. 39-43, the user query specifies multiple join operations, and statistics are generated for two or more join operations and used to determine an order in which said join operations are performed, e.g. the join operation with the lowest statistic is performed first. Hence, Amundsen suggests generating a first portion of a result set for the query while running the query according to the first join order, and further suggests generating a second portion of the result set for the query while running the query according to the preferred join order. That is, as Amundsen states, it is most efficient, generally speaking, to perform first those parts of the query that produce the smallest solution set of tuples (i.e. generating a first portion of a result set for the query while running the query according to the first join order), and because Amundsen discloses a multi-join operations of course, generating a second portion of the result set for the query while running the query according to the preferred join order is further suggested.

Both Amundsen's and Zait's system disclose systems that further optimize queries, and therefore are both within the same field of endeavor. For the above reasons, it would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Amundsen's teachings of performing first those parts of the query that produce the smallest solution set of the tuples to Zait's system for minimizing the number of intermediate results that must be produced, stored, and then processed in later parts of the query.

Claim 13:

As to claim 13, Zait and Amundsen disclose the claimed limitation of claim 12, Amundsen further disclosing "determining respective fan-in statistics for each of the join operations" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.); "and selecting the preferred join order based on the fan-in statistics" (col. 13 lines 40-43, multiple join operations. Join operation with the lowest statistic is performed first.).

Claim 14:

As to claim 14, Zait and Amundsen disclose the claimed limitation of claim 12, Amundsen further disclosing "determining respective fan-out statistics for each of the join operations" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.); and "selecting the preferred join order based on the fan-out statistics" (col. 13 lines 40-43, multiple join operations. Join operation with the lowest statistic is performed first.).

Claim 15:

As to claim 15, Zait and Amundsen disclose the claimed limitation of claim 12, Zait further disclosing "performing the step of selecting a preferred join order after collecting statistics for a predetermined number of records from a table involved in the query" [0034, alter execution plan].

Claim 16:

Zait discloses "an apparatus for executing a query comprising:"

"at least one processor"[0074, one or more processors];

"a memory coupled with the at least one processor" [0074, processors execute instructions contained in memory]; and

"a database engine residing in the memory and executed by the at least one processor, the database engine configured to run a query involving a plurality of join operations according to a first join order" [0023-0024, 0023, query statements run by database system. 0024, execution plan is run which contains one or more operations; and therefore a plurality of statistics is collected about the operations. Operations may include join operations.];

"concurrent with running the query, collect statistics about each of the join operations" [0024, statistics information is collected from operation which may include join operations];

"select a preferred join order, while running the query, based on the collected statistics, such that the query continues to run according to the preferred join order" (0034, The plurality of collected execution statistics are then used to improve performance of the query statement can be improve the performance of the query statement. Performance of the query statement may include altering the execution plan (i.e. changing a first join order to a preferred join order based on statistics is possible.).

Zait does not explicitly disclose "generate a first portion of a result set for the query while running the query according to the first join order" and "generate a second portion of the result set for the query while running the query according to the preferred join order"

On the other hand, Amundsen discloses if a multi-way join is requested by the user, e.g. a first relation is to be joined on a given attribute with a second relations to be joined on a given attribute with a third relation, it is useful to know the join fanout (i.e. number of intermediate results that will be produced by, the join of the first and second relations as compared to the join fanout of the join of the second and third relations) (col. 23 lines 17-21). When processing a compound sql query, it is most efficient, generally speaking, to perform first those parts of the query that produce the smallest solution set of tuples, because doing so minimizes the number of intermediate results that must be produced, stored , and then processed in later parts of the query (col. 23 lines 25-30). Col. 13 l. 39-43, the user query specifies multiple join operations, and statistics are generated for two or more join operations and used to determine an order in which said join operations are performed, e.g. the join operation with the lowest statistic is performed first. Hence, Amundsen suggests to generate a first portion of a result set for the query while running the query according to the first join order, and further suggests to generate a second portion of the result set for the query while running the query according to the preferred join order. That is, as Amundsen states, it is most efficient, generally speaking, to perform first those parts of the query that produce the smallest solution set of tuples (i.e. generate a first portion of a result set for the query while running the query according to the first join order), and because

Amundsen discloses a multi-join operations of course, generating a second portion of the result set for the query while running the query according to the preferred join order is further suggested.

Both Amundsen's and Zait's system disclose systems that further optimize queries, and therefore are both within the same field of endeavor. For the above reasons, it would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Amundsen's teachings of performing first those parts of the query that produce the smallest solution set of the tuples to Zait's system for minimizing the number of intermediate results that must be produced, stored, and then processed in later parts of the query.

Claim 18:

As to claim 18, Zait and Amundsen disclose the claimed limitation of claim 12, Zait further disclosing

"a manager configured to select the preferred join order" [0024, execution plan is generated. 0035, Executor and optimizer manage operations];

"an execution engine, coupled with the manager, and configured to execute the query according to the preferred join order" [0024, selection operation is executed]; and

"a statistics collector, coupled with the manager and the execution engine, configured to monitor execution of each of the plurality of join operations, capture respective performance data" [0024, collection of executed statistics]; and "communicate the respective performance data to the manager" [0035, changes execution plan based on statistics data].

Claim 19:

As to claim 19, Zait and Amundsen disclose the claimed limitation of claim 12, Amundsen further disclosing "wherein the statistics include respective fan-in statistics and respective fan-out statistics for each of the join operations" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.).

Claim 20:

As to claim 20, Zait and Amundsen disclose the claimed limitation of claim 12, Zait further disclosing "wherein the statistics are collected for a predetermined number of records from a table involved in the query" [0032].

Claim 21:

Zait discloses "a program product, comprising:"

"program code configured upon execution to perform the steps of: "

"running the query according to a first join order;" [0024 and 0003, query is executed according to the one or more operations in an execution plan. Operations may consist of a join order. 0004, discloses determining an optimal join order therefore there must exist other join orders and the most optimal one is selected.]

"concurrent with running the query, collecting statistics about each of the join operations" [0024, statistics information is collected from operation which may include join operations],

"based on the collected statistics, selecting a preferred join order, while running the query, such that the query continues to run according to the preferred join order" (0034, The plurality of collected execution statistics are then used to improve performance of the query statement can be improve the performance of the query statement. Performance of the query statement may include altering the execution plan (i.e. changing a first join order to a preferred join order based on statistics is possible.));

"generating a second portion of the result set for the query while running the query according to the preferred join order"; and

"a tangible computer readable medium bearing the program code".

Zait does not explicitly disclose "generating a first portion of a result set for the query while running the query according to the first join order" and "generating a second portion of the result set for the query while running the query according to the preferred join order"

On the other hand, Amundsen discloses if a multi-way join is requested by the user, e.g. a first relation is to be joined on a given attribute with a second relations to be joined on a given attribute with a third relation, it is useful to know the join fanout (i.e. number of intermediate results that will be produced by, the join of the first and second relations as compared to the join fanout of the join of the second and third relations) (col. 23 lines 17-21). When processing a compound sql query, it is most efficient, generally speaking, to perform first those parts of the query that produce the smallest solution set of tuples, because doing so minimizes the number of intermediate results that must be produced, stored , and then processed in later parts of the query (col. 23 lines 25-30). Col. 13 l. 39-43, the user query specifies multiple join operations, and statistics are generated for two or more join operations and used to determine an order in which said join operations are performed, e.g. the join operation with the lowest statistic is performed first. Hence, Amundsen suggests generating a first portion of a result set for the query while running the query according to the first join order, and further suggests generating a second portion of the result set for the query while running the query according to the preferred join order. That is, as Amundsen states, it is most efficient, generally speaking, to perform first those parts of the query that produce the smallest solution set of tuples (i.e. generating a first portion of a result set for the query while running the query according to the first join order), and because Amundsen discloses a multi-join operations of course, generating a second portion of the result set for the query while running the query according to the preferred join order is further suggested.

Both Amundsen's and Zait's system disclose systems that further optimize queries, and therefore are both within the same field of endeavor. For the above reasons, it would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Amundsen's teachings of performing first those parts of the query that produce the smallest solution set of the tuples to Zait's system for minimizing the number of intermediate results that must be produced, stored, and then processed in later parts of the query.

Claim 22:

As to claim 22, Zait and Amundsen disclose the claimed limitation of claim 21, Amundsen further disclosing "wherein the program code is further configured to: determining respective fan-in statistics for each of the join operations" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.) and "selecting the preferred join order based on the fan-in statistics" (col. 13 lines 40-43, multiple join operations. Join operation with the lowest statistic is performed first.)

Claim 23:

As to claim 22, Zait and Amundsen disclose the claimed limitation of claim 21, Amundsen further disclosing "wherein the program code is further configured to: determining respective fan-out statistics for each of the join operations" (col. 23 lines 21-25, it is useful to know the join fanout, i.e., number of intermediate results that will be produced by the join of the first and second relations as compared to the join fanout of the join of the second and third relations.) and "selecting the preferred join order based on the fan-out statistics" (col. 13 lines 40-43, multiple join operations. Join operation with the lowest statistic is performed first.).

Response to Arguments

4. Applicant's arguments with respect to claims 1,3-16, and 18-23 have been considered but are moot in view of the new ground(s) of rejection.

Applicant's assert the following (lettered):

A. Applicant's response (hereafter Response) page 9, that there is no disclosure or suggestion in these passages, however that the updates to the execution plan used by a query statement are used in the same execution of the query statement during which statistics are collected. Further asserting that indeed block 710 of fig. 7, which refers to improving the performance of the query statement is performed just before the end of the flowchart, and does not refer to continuing the query. That Zait does nothing to address any sub-optimal performance of a query that is currently being executed.

In reply, the examiner respectfully disagrees with Applicant's. As stated in 0034, an execution plan is generated for **a query statement** (702). The execution plan includes one or more operations. One of the one or more operations in the execution plan is selected (704). The selected operation is executed (706) and a plurality of execution statistics are then **used** to improve performance of **the query statement** (710). Hence, it appears although 710 is performed just before the end of the flowchart, and does not refer to continuing to run the query, it appears that Zait still suggests it within the specifications stating that the statistics are used to improve the stated query statement (hence that same query).

Even if Zait did not disclose the following, Amundsen further discloses the assertion by utilizing the statistics generated within the same query. Therefore, Applicant's assertions are unpersuasive over the cited references.

B. Response page 10, That Zait does not disclose first and second portions of the same result set, for the same execution query, are generated before and after a dynamic change in join order.

In reply, the argument is moot on grounds of new rejection.

C. Response page 10, Iyer does not disclose either collecting statistics during execution of a query or dynamically changing a join order during the execution of a query such that first and second portions of a result set are generated using different join orders.

In reply, the argument is moot on grounds of new rejection.

D. Response page 10, that Zait nor Iyer appreciates that join order may be selected dynamically during execution of a query based upon statistics collected during the same execution of that query, and that different portions of the same result set can be generated during the execution of a query using different join orders.

In reply, the argument is moot on grounds of new rejection.

E. Response page 10-11, that Zait discloses generating a second execution plan in paragraph 0035, however the passage does not disclose that the second execution plan is used to continue the processing of the same execution query.

In reply, the examiner respectfully disagrees. Please see above section A.

F. Response page 11, that Zait figure 8 does not disclose or suggest that any change may occur in the execution plan used by executor 812 during one execution of the query statement. That the figure does not disclose or suggest that different portions of the same result set for a query can be generated using different execution plans.

In reply, the argument is moot on grounds of new rejection.

G. Response page 11-12, Claim 12, 16, and 21 are similarly amended as claim 1. Claims dependent to independent claims are allowable based on dependency.

In reply, the examiner respectfully disagrees. Claims 12, 16, and 21 are similarly rejected like claim 1. Claims that depend from independent claims are therefore further rejected.

Conclusion

5. The prior art made of record listed on PTO-892, and not relied upon, if any, is considered pertinent to applicant's disclosure.

Contact Information

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael D. Pham whose telephone number is (571) -272-3924 or fax (571) – 273 - 3924. The examiner can normally be reached on Monday - Friday 8am - 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on (571) - 272-4049. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Pham
Art Unit 2167
Examiner M.P.

Cam Y Truong
Art Unit 2162
Primary Examiner
cy

John Cottingham
Art Unit 2167
Supervisor

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100